REMARKS

Claims 1, 9, 12, and 13 have been amended. Claims 7 and 15 have been cancelled. Claims 1-6, 8-14, and 16 remain pending in the application for consideration. In view of the following remarks, Applicant respectfully requests reconsideration and allowance of the subject application.

§103 Rejection

Claims 1-5, 8-13, and 16 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 7,107,548 to Shafron (hereinafter "Shafron") in view of U.S. Patent 6,266,805 to Nwana et al. (hereinafter "Nwana").

Claims 6, 7, 14, and 15 are rejected under 35 U.S.C. §103(a) as being unpatentable over Shafron, in view of Nwana, and further in view of U.S. Patent Application 2002/0196279 to Bloomfield et al. (hereinafter "Bloomfield"). Applicant respectfully traverses these rejections.

Before discussing the specific rejections in the current Office Action, the following discussion of Shafron, Nwana, and Bloomfield is provided.

The Shafron Reference

Shafron is directed to controlling an Internet browser. Specifically, a user may automatically receive information such as stock quotes, email, or news

headlines while using the Internet browser. The Internet browser is customized using a software provided by an Internet content provider, an Internet Service Provider (ISP), or software that resides on the Internet user's computer.

The Nwana Reference

Nwana is directed to a visualization apparatus for software systems that is used to control, monitor, or manage a process. The software system includes a plurality of software modules which communicate with one another. The visualization apparatus includes means for storing and displaying communications between the software modules, means for obtaining the software modules organizational data, means for processing the software modules communications, and means for displaying the software modules communications in response to an organization or communications. The software modules are debugged or analyzed by visualizing the communications between them.

The Bloomfield Reference

Bloomfield is directed to displaying application output data within windows embedded in a web browser. The windows can be moved, resized, and manipulated within the browser window even when the application program is non-web enabled (*i.e.*, a legacy application). The application programs can reside

on different servers and data from the different servers is displayed within windows in the same web browser window.

The Claims

Claim 1 has been amended and recites a method, comprising [added language is indicated in bold italics]:

- generating a user interface that identifies add-ons associated with an application program, the user interface comprising
    - a plurality of lists of add-ons, a status of each listed add-on, a disable add-on function, and an enable add-on function,
    - the plurality of lists of add-ons comprising a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program,
    - ***wherein the user interface identifies add-ons by displaying the name of a add-on, a publisher of the add-on, a status of the add-on, a type of add-on, and a time when the add-on was last accessed***; and
- in response to user input, managing the enable/disable state of said add-ons by:
    - determining if the user has selected a list of add-ons from the plurality of lists of add-ons:
    - if the user has selected a list of add-ons, displaying the selected list of add-ons;
    - determining if the user has selected a particular add-on from the displayed list of add-ons;
    - if the user has selected a particular add-on from the displayed list of add-ons, determining if the user has chosen to disable or enable the particular add-on by activating the disable add-on or enable add-on function of the user interface;
    - if the user has chosen to disable the particular add-on by activating the disable add-on function of the user interface, disabling the add-on; and
    - if the user has chosen to enable the add-on by activating the

enable add-on function of the user interface, enabling the add-on.

Applicant has cancelled Claim 7 without prejudice and incorporated its elements into Claim 1. Accordingly, "wherein the user interface identifies add-ons by displaying the name of the add-on, the publisher of the add-on, the status of the add-on, the type of add-on, and the time when the add-on was last accessed" is supported by the Applicant's Specification.

Shafron and Nwana fail to disclose, teach, or suggest "the plurality of lists of add-ons comprising a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program'', as recited by Claim 1.

In making out the rejection of Claim 1, the Offices conceded that "Shafron fails to specifically show the recited element" (Office Action Page 3).

The Office then cites to Column 37, lines 48-64 of Nwana for disclosing the cited element. Nwana discloses that "the reports tool provides a global view of problem solving in a society of agents and is useful both as a ***debugging and an administrative tool***. It allows a user to select a set of agents and request that they report the status of all their jobs to it. Next, the user can select an agent of interest and a job owned by that agent... For the selection of agent and job, ***the reports tool generates the GANTT chart*** type graph 700 of Fig. 7

showing the decomposition 560 of the job, the allocation of its constituent subparts to different agents in the community, and the relevant states of the job and subparts. Other attributes of the job might also be shown on the chart, such as when each agent is scheduled to perform its part, their costs, the priority assigned, and the resources required".

In other words, Nwana employs a GANTT chart to debug and administer agents. The GANTT displays a decomposition of a job, the allocation of the jobs subparts, and the status of the job and subparts. There is nothing in the cited section, as well as the entire Nwana reference, that discloses, teaches or suggests a plurality of lists of add-ons comprising "a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program''. The recited element is simply missing.

Accordingly, Shafron and Nwana fail to disclose, teach, or suggest "a plurality of lists of add-ons comprising "a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program" as recited in Claim 1.

Shafron, Nwana, and Bloomfield fail to disclose, teach, or suggest "wherein the user interface identifies add-ons by displaying a name of the add-

on, a publisher of the add-on, a status of the add-on, a type of add-on, and a time when the add-on was last accessed", as recited by Claim 1.

In making out the rejection of Claim 7 whose subject matter is now incorporated in Claim 1, the Office conceded that "Shafron and Nwana do not explicitly disclose the recited element" (Office Action Page 8). The Office then cites to Paragraphs 0039 and 0044 of Bloomfield for disclosing recited the element.

Bloomfield discloses "[t]he data exchanged between the client agent 136 and server agent 134 during the client-server session includes not only input events and the graphical output data of the application program 130, but also ***window attribute information (e.g., window position, z-order, size, style, color etc.)***. The window attribute information of the application output windows 326 is initially specified by the application objects 128 embedded in the web page 322. ***For example, the application objects 128 can include an ActiveX control which specifies and controls the window attributes of the application output windows 326 during the client-server session.*** In one embodiment, the application-output windows 326 exhibit the same dimensions as the corresponding ActiveX controls" (Paragraph 0039).

Bloomfield further discloses "[i]n one embodiment, ***each input event affecting the application output window 326 is transferred to and processed by the server agent 114, which then instructs the client agent 136 to effect***

corresponding changes in the application output window 326. In another embodiment, one or more input types (*e.g.*, mouse actions) are processed entirely by the client agent 136 and not reported to the server agent 134, where the graphical output data displayed within the application output window 326 remains unchanged" (Paragraph 0044).

Contrary to the Office's assertion, Bloomfield is directed to controlling the attributes (*i.e.*, window position, z-order, size, style, color etc.) of an application's output window. Neither the cited paragraphs, nor the entire Bloomfield reference, discloses, teaches, or suggests "wherein the user interface identifies add-ons by displaying the name of the add-on, the publisher of the add-on, the status of the add-on, the type of add-on, and the time when the add-on was last accessed". The recited element is simply missing.

Accordingly, Shafron, Nwana, and Bloomfield fail to disclose, teach, or suggest "wherein the user interface identifies add-ons by displaying the name of the add-on, the publisher of the add-on, the status of the add-on, the type of add-on, and the time when the add-on was last accessed" as recited in Claim 1.

Accordingly, for at least these reasons, this claim is allowable and the Office's rejection is traversed.

Claims 2-6 and 8 depend from independent Claim 1 and are allowable by virtue of their dependency from Claim 1, as well as for the additional features that

the claims recite.

Claim 9 has been amended and recites a computer readable storage medium comprising computer-executable instructions that, when executed by a processor, perform a method comprising [added language is indicated in bold italics]:

- generating a user interface, the user interface configured to identify add-ons associated with an application program,
    - the user interface comprising a plurality of lists of add-ons, a status of each listed add-on, a disable add-on function, and an enable add-on function,
    - the plurality of lists of add-ons comprising a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program,
    - ***wherein the user interface identifies add-ons by displaying a name of the add-on, a publisher of the add-on, a status of the add-on, a type of add-on, and a time when the add-on was last accessed***; and
- in response to a user input, manages the enable/disable state of said add-ons by:
    - determining if the user has selected a list of add-ons from the plurality of lists of add-ons;
    - if the user has selected a list of add-ons, displaying the selected list of add-ons:
    - determining if the user has selected a particular add-on from the displayed list of add-ons;
    - if the user has selected a particular add-on from the displayed list of add-ons, determining if the user has chosen to disable or enable the particular add-on by activating the disable add-on or enable add-on function of the user interface;
    - if the user has chosen to disable the particular add-on lay activating the disable add-on function of the user interface, disabling the add-on; and
    - if the user has chosen to enable the add-on by activating the enable add-on function of the user interface, enabling the add-

on.

For reasons similar to those made in regard to Claim 1, Shafron and Nwana fail to disclose, teach, or suggest "the plurality of lists of add-ons comprising a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program" as recited in Claim 9.

As discussed above, the GANTT disclosed by Nwana displays a decomposition of a job, the allocation of the jobs subparts, and the status of the job and subparts. There is nothing in the cited section, as well as the entire Nwana reference, that discloses, teaches or suggests a plurality of lists of add-ons comprising "a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program''.

Accordingly, Shafron and Nwana fail to disclose, teach, or suggest a "the plurality of lists of add-ons comprising a list of add-ons that have been used by the application program, a list of add-ons that are currently used by the application program, and a list of add-ons that are currently blocked by the application program" as recited in Claim 9.

For reasons similar to those made in regard to Claim 1, Shafron, Nwana, and Bloomfield fail to disclose, teach, or suggest "wherein the user

interface identifies add-ons by displaying a name of the add-on, a publisher of the add-on, a status of the add-on, a type of add-on, and a time when the add-on was last accessed" as recited in Claim 9.

As discussed above, Bloomfield is directed to controlling the attributes (*i.e.*, window position, z-order, size, style, color etc.) of an application's output window. Neither the cited paragraphs, nor the entire Bloomfield reference, discloses, teaches, or suggests "wherein the user interface identifies add-ons by displaying a name of the add-on, a publisher of the add-on, a status of the add-on, a type of add-on, and a time when the add-on was last accessed".

Accordingly, Shafron, Nwana, and Bloomfield fail to disclose, teach, or suggest "wherein the user interface identifies add-ons by displaying a name of the add-on, a publisher of the add-on, a status of the add-on, the type of add-on, and a time when the add-on was last accessed" as recited in Claim 9.

Claims 10-14 and 16 depend from independent Claim 9 and are allowable by virtue of their dependency from Claim 9, as well as for the additional features that the claims recite.

Conclusion

Applicant respectfully submits that Claims 1-6, 8-14, and 16 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. Should any matter remain unresolved, the undersigned respectfully requests a telephone conference with the Examiner to resolve any outstanding matter.

Respectfully Submitted,

Dated: 2/5/09

By: _____

Jeffrey Valley
Reg. No. 50,653
(509) 755-7236